Mini Project on HOME AUTOMATION SYSTEM

Group : **B10**

204239 - Madu Varun Reddy
204240 - Mallipudi Harshadeep
204241 - Manav Bhadoria
204242 - Manda Abhinay Reddy

Faculty :

Dr. Prithvi Pothupogu
Dr. T V K Hanumantha Rao

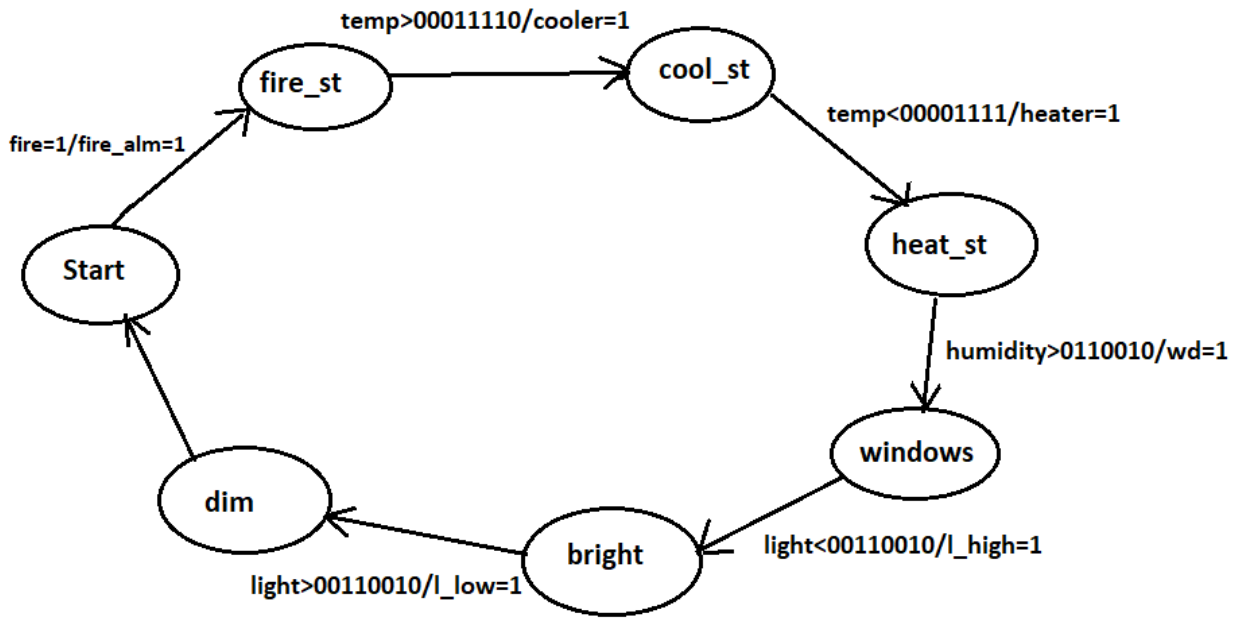
Department of **Electronics and Communication Engineering**

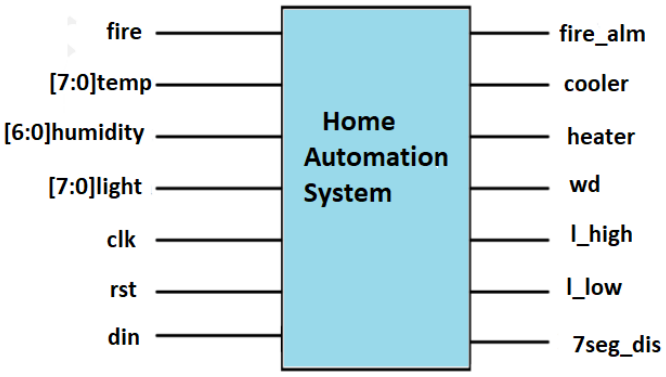
Year of Submission : 2022

Abstract :

This project aims to automate home systems by using digital system design. Home automation involves introducing a degree of computerized or automatic control over certain electrical and electronic systems in a home. Our project aims to increase both security and comfort without human interference. Instead we automate these processes using digital system design and Verilog HDL. This device has been modeled such that it takes care of 4 factors namely: **fire, temperature, humidity, and luminosity**. Our project is installed in a home to control and monitor the various appliances. Based on the configuration of the hardware system, the system controls the appliances. The hardware system will be composed of four sensors for the processes we have chosen to automate. The sensors used in the hardware system are a smoke detector, optical sensitive devices, a thermostat, and a humidity sensor. The solution uses the Digital System Design concepts of a state machine to design a Mealy system that is simulated using Verilog HDL in Xilinx. A sequential pattern of controlling the four sensors is followed in a priority order.

State Flow Diagram :





Inputs :

fire - signal from smoke detector ('1' if smoke is detected)
temp - signal from thermostat
humidity - signal from humidity sensor
light - luminosity
clk - clock
rst - reset
din - data in

Outputs :

fire_alm - fire alarm (buzzes if fire is HIGH ,i.e '1')
cooler - if temp is greater than 30 degrees Celsius , then it turns ON Air-Conditioner
heater - if temp is lesser than 15 degrees Celsius , then it turns ON Heater
wd - if humidity is greater than 50%, opens the windows
l_high - if luminosity is less than 50 lux , then increase the intensity of light
l_low - if luminosity is greater than 50 lux , then decrease the intensity of light
7seg_dis - 7 Segment display

Theory :

- **About Verilog HDL:**

  Our project is implemented using Verilog HDL in Xilinx. A Hardware Description Language(HDL) is a computer-based language that describes the hardware of digital systems in a textual form. An HDL is a modeling language rather than a computational language. An HDL resembles an ordinary computer programming language, such as C, but is specifically oriented to describing hardware structures and behavior of a digital design. Verilog is a hardware description language used to model electronic systems. It is most commonly used in the design and verification of digital circuits at the register transfer level of abstraction. Verilog differs from traditional software languages because they include ways of describing the

propagation of time and signal dependencies (sensitivity). A Verilog design consists of a hierarchy of modules. Modules encapsulate design hierarchy and communicate with other modules through a set of declared input, outputs, and bidirectional ports.There are two assignment operators, a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state machine update without needing to declare and use temporary storage variables.Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. At the time of Verilog's introduction (1984), Verilog represented a tremendous productivity improvement for circuit designers who were already using graphical schematic capture software and specially written software programs to document and simulate electronic circuits.

Designs, which are described in HDL are independent of technology, very easy for designing and debugging, and are normally more useful than schematics, particularly for large circuits.Verilog supports a design at many levels of abstraction.Those majorly include Behavioral, Structural, Data-flow style of modeling.

Behavioral Style of Modeling : This level describes a system by concurrent algorithms (Behavioral). Every algorithm is sequential, which means it consists of a set of instructions that are executed one by one. Functions, tasks and blocks are the main elements. There is no regard to the structural realization of the design.

Structural Style of Modeling : The structural model describes a system using basic components such as digital gates and adders. In structural modeling, the programmer or the designer thinks about the circuit as a box or a module. It is encapsulated from the outer environment. In other words, it communicates with the outer environment through inputs and outputs.Moreover, it is possible to describe the structure inside a module using gates and submodules. Also, it defines how these modules are connected to each other and to the module ports. Furthermore, the structural model helps to draw a schematic diagram for the circuit.

Data-flow Style of Modeling: In this we use “assign” statement more frequently. We use logical operators such as |,&,^,~ etc.

- **Home Automation** :

Home automation systems involve introducing a degree of automation to electrical appliances we use in our day-to-day life to increase both comfort and security without human interference. Utilizing home automation could lead to more efficient and intelligent energy saving techniques. Home Automation allows smart and intelligent control of home appliances using Digital System Design. Home automation systems can control many things in a home, which are precious and valuable to a person. For example, the doors, windows, fire alarm and temperature can be controlled by just one click, just to ensure that all these are safe, when the occupants of the house are either at home or out or to just sit in one place and control these devices instead of having to get to that place to perform a certain action. The controller does its work of checking each of the devices periodically and in a certain order. The system can be controlled by a remote or any device that can be connected to all the devices that need to be secure. Their communication is through sensors which are attached on all the devices. This is a relatively new concept and has not hit the market so extensively yet, but once it does then it will definitely be very much in demand and so there will be a constant need for something so secure and safe. The era of automation began with the sensors and it's respective outputs such as automatic opening and closing of the door or turning on and off of the street lights based on the intensity of the light. This idea was further modified to control home lighting system and fan based on intensity of the light and temperature respectively.The rapid growth of technologies influence us to use smartphones to remotely control the home appliances. An automated devices has ability to work with versatility, diligence and with lowest error rate. The idea of home automation system is a significant issue for Researchers and home appliances companies. Automation system not only helps to decrease the human labor but it also saves time and energy. Early home automation systems were used in labor saving machines but nowadays its main objective is provide facilities to elderly and handicapped people to perform their daily routine tasks and control the home appliances remotely

This device has been modeled using Verilog HDL such that it takes care of 4 aspects of a home namely fire, temperature, humidity, and luminosity. A sequential pattern of controlling fire, temperature, humidity, and luminosity is followed in priority order. Our project uses system design concepts of a state machine to design a mealy system. The objective of this project is to design a controller which provides an automated home system. The parameter sensors connected to the controller will provide the required signals that will activate the controller processes and take the specified actions required based on the inputs. We modeled our design mostly using Behavioral Style of Modeling.

Our control system is a mealy machine since it is a finite state machine whose output values are determined by both its current state and the current inputs. There are sensors for all the four processes we aim to automate. The sensors used in the hardware system are a smoke detector, optical sensitive devices, a thermostat, and a humidity sensor. The priority order of the sensors is a smoke detector, thermostat, humidity sensor, and optical sensor device. The highest priority is the smoke detector since it can have massive health and safety hazards.

- **States Flow :**

Initially all the states are assigned with numbers. The states are “Start “, “fire_st”, “cool_st”, “heat_st”, “windows”, “bright”, “dim” . The first state is the start state, in this state all the outputs are set to 0.

Fire Alarm State:

The sensor in the fire alarm state is the smoke detector. If it detects smoke then the input it sends is 1 or else it remains 0 as it was in the start state. If the input is 1 then the fire alarm is set to HIGH(i.e 1) and starts ringing till it is manually switched off. The next state is Temperature Control State

Temperature Control State:

The sensor in this state is the thermostat, it has 3 conditions it has to fulfill depending on the input from the thermostat if the temperature is

above 30 degrees Celsius then it turns on the air conditioner, the second condition is if the input received from the thermostat is lesser than 15 degrees if this is the case then the heater is turned on. Finally, the last condition is the start state where the thermostat keeps checking the temperature of the room in case it drops below 15 degrees or rises above 30 degrees. The next state is Humidity Control State

Humidity Control State: The sensor in this state is the humidity sensor, it has two conditions to fulfill depending on the input it receives. If the humidity is greater than 50 % then open the windows. The specifications for this state are humidity>0110010. If the humidity isn't above 50 percent then keep the system in the first state. The next state is Luminosity Control State.

Luminosity Control State: This state has two conditions to fulfill depending on the input it receives from the light-dependent resistor which keeps checking the light intensity in the room. If the light intensity goes below 50 lux or light < 110010 then increase the current through the light source and if the light intensity goes above 50 lux or if light > 110010 then decrease the current through the light source.

The state diagram(as shown in Abstract) step is the next to follow in which all the states are shown and also see if the states can be reduced. A state diagram is developed that shows all the states and their behaviors. The states are examined for redundancy and reduced if any are found. This analysis and reduction helps to obtain a minimum and race free system. Simulate the system using Verilog HDL and see if it is working efficiently. Based on the results obtained, the results are compared and verified with those that were assumed and hence the conclusions are obtained. This report gives an in-depth knowledge of the applied project and its functionality along with the limitations, advantages, conclusions, future development and other factors. The report is divided into chapters and each chapter deals with one part of the applied project

Code :

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 17.04.2022 23:28:15
// Design Name:
// Module Name: homeAutomation
// Project Name: Home Automation System
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

// assigning numbers to states
`define start 4'd0
`define fire_st 4'd2
`define cool_st 4'd3
`define heat_st 4'd4
`define windows 4'd5
`define bright 4'd6
`define dim 4'd7
`define comb_lock 4'd8
```

```
module
homeAutomation(clk,reset,din,fire,temp,humidity,light,seg_out,fire_alm,heater,cooler,wd,
l_high,l_low);
input clk,reset,fire,din;
input [7:0] temp,light;
input [6:0] humidity;
output reg fire_alm,heater,cooler,wd,l_high,l_low;
output [7:0] seg_out;
reg [3:0] current_st;
reg [3:0] next_st;
wire dclk;
wire [3:0] bcd_data;
debounce_filter m1 (.clk(clk), .rst (reset), .din(din), .dout(dclk));
bcd_counter m2 (.clk(dclk), .rst(reset), .q (bcd_data));
bcd_7seg m3 (.bcd_in(bcd_data), .sseg(seg_out));
initial
begin
   current_st=`start;
   next_st= `start;
   fire_alm='b0;
   cooler='b0;
   heater='b0;
   wd='b0;
   l_high='b0;
   l_low='b0;
end

always @(posedge clk)
current_st=next_st;
always @(current_st)
begin
   case(current_st)
   `start:
   begin
      fire_alm='b0;
      cooler='b0;
      heater='b0;
      wd='b0;
      l_high='b0;
      l_low='b0;
```

```
    end
    `fire_st: fire_alm='b1;
    `cool_st: cooler='b1;
    `heat_st: heater='b1;
    `windows:wd='b1;
    `bright: l_high='b1;
    `dim: l_low='b1;
    endcase
end
always @(current_st,fire,temp,humidity,light,reset,din)
begin
    if(reset=='b1)
        next_st=`start;
    else
        case(current_st)
        `start:
            begin
            if(fire=='b1)
            next_st=`fire_st;
            else if(temp > 'b00011110)
            next_st=`cool_st;
            else if(temp < 'b00001111)
            next_st=`heat_st;
            else if(humidity>'b0110010)
            next_st=`windows;
            else if(light < 'b00110010)
            next_st=`bright;
            else if(light > 'b00110010)
            next_st=`dim;
            else next_st=`start;
            end
         `fire_st:

            begin
            if(temp > 'b00011110)
            next_st=`cool_st;
            else if(temp < 'b00001111)
            next_st=`heat_st;
            else if(humidity>'b0110010)
            next_st=`windows;
```

```
    else if(light < 'b00110010)
    next_st=`bright;
    else if(light > 'b00110010)
    next_st=`dim;
    else next_st=`start;
    end
  `cool_st:
    begin
    if(temp < 'b00001111)
    next_st=`heat_st;
    else if(humidity>'b0110010)
    next_st=`windows;
    else if(light < 'b00110010)
    next_st=`bright;
    else if(light > 'b00110010)
    next_st=`dim;
    else next_st=`start;
    end
  `heat_st:
    begin
    if(humidity>'b0110010)
    next_st=`windows;
    else if(light < 'b00110010)
    next_st=`bright;
    else if(light > 'b00110010)
    next_st=`dim;
    else next_st=`start;
    end
  `windows:
    begin
    if(light < 'b00110010)
    next_st=`bright;
    else if(light > 'b00110010)
    next_st=`dim;
    else next_st=`start;
    end
  `bright:
    begin
    if(light > 'b00110010)
    next_st=`dim;
```

```
            else next_st=`start;
            end
          `dim:
            begin
            next_st=`start;
            end
          endcase
end
endmodule

//Debounce Filter
module debounce_filter(
 input din,
 input clk,
 input rst,
 output dout
 );
reg [2:0] temp;
always @ (posedge clk or posedge rst)
begin
    if (rst == 1'b1)
        temp <= 4'b000;
    else
        temp <= {temp[1:0], din};
end
assign dout = temp[0] & temp[1] & (!temp[2]);
endmodule

//BCD Counter
module bcd_counter(
 input clk,
 input rst,
 output reg [3:0] q
 );
always@ (posedge clk or posedge rst)
begin
    if (rst)
        q <= 4'b0000;
    else if (q == 4'b1001)
        q <= 4'b0000;
```

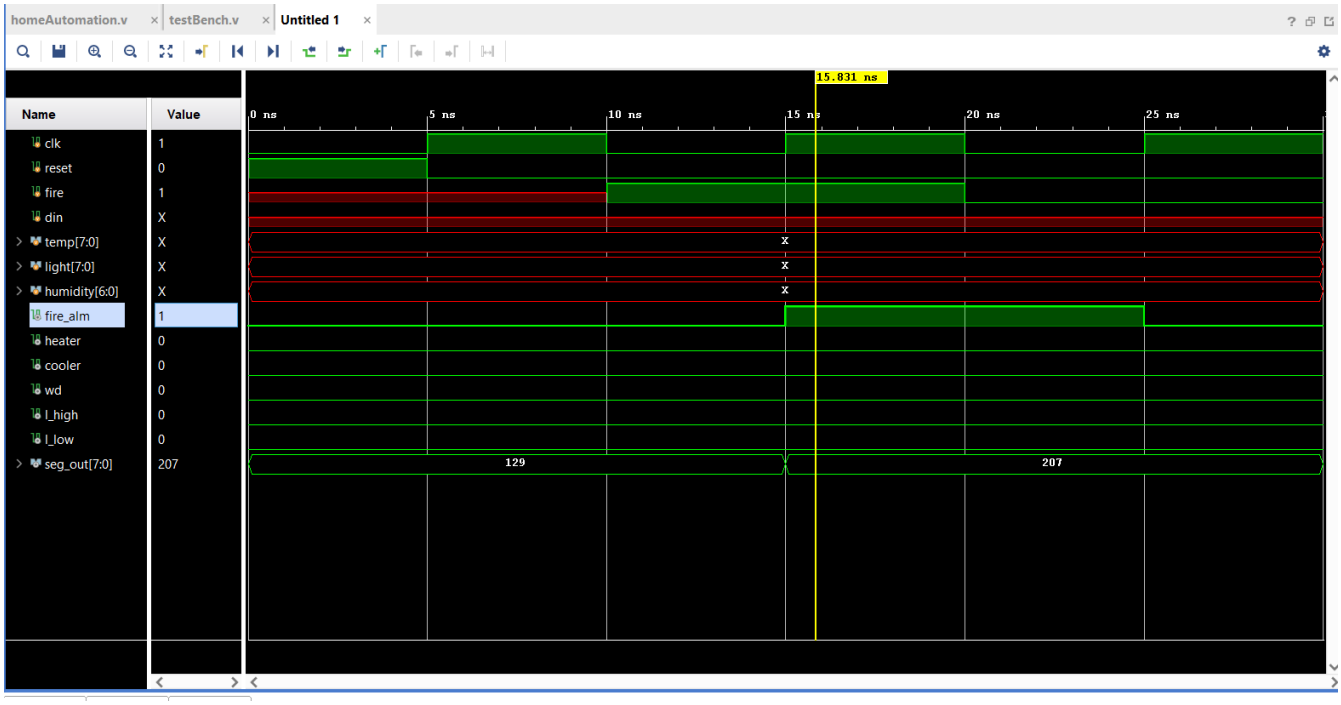
```
    else
        q <= q + 1;
end
endmodule

//BCD to 7 Segment Display
module bcd_7seg(
 input wire [3:0] bcd_in,
 output reg [7:0] sseg
 );
always @ (bcd_in)
begin
    case (bcd_in)
    4'b0000:
        sseg = 8'b10000001; //01111110;
    4'b0001:
        sseg = 8'b11001111; //00110000;
    4'b0010:
        sseg = 8'b10010010; //01101101;
    4'b0011:
        sseg = 8'b10000110; //01111001;
    4'b0100:
        sseg = 8'b11001100 ;//00110011;
    4'b0101:
        sseg = 8'b10100100; //01011011;
    4'b0110:
        sseg = 8'b10100000; //01011111;
    4'b0111:
        sseg = 8'b10001111; //01110000;
    4'b1000:
        sseg = 8'b10000000; //01111111;
    default : sseg = 8'b10000100; //01111011;
    endcase
end
endmodule
```

Results :

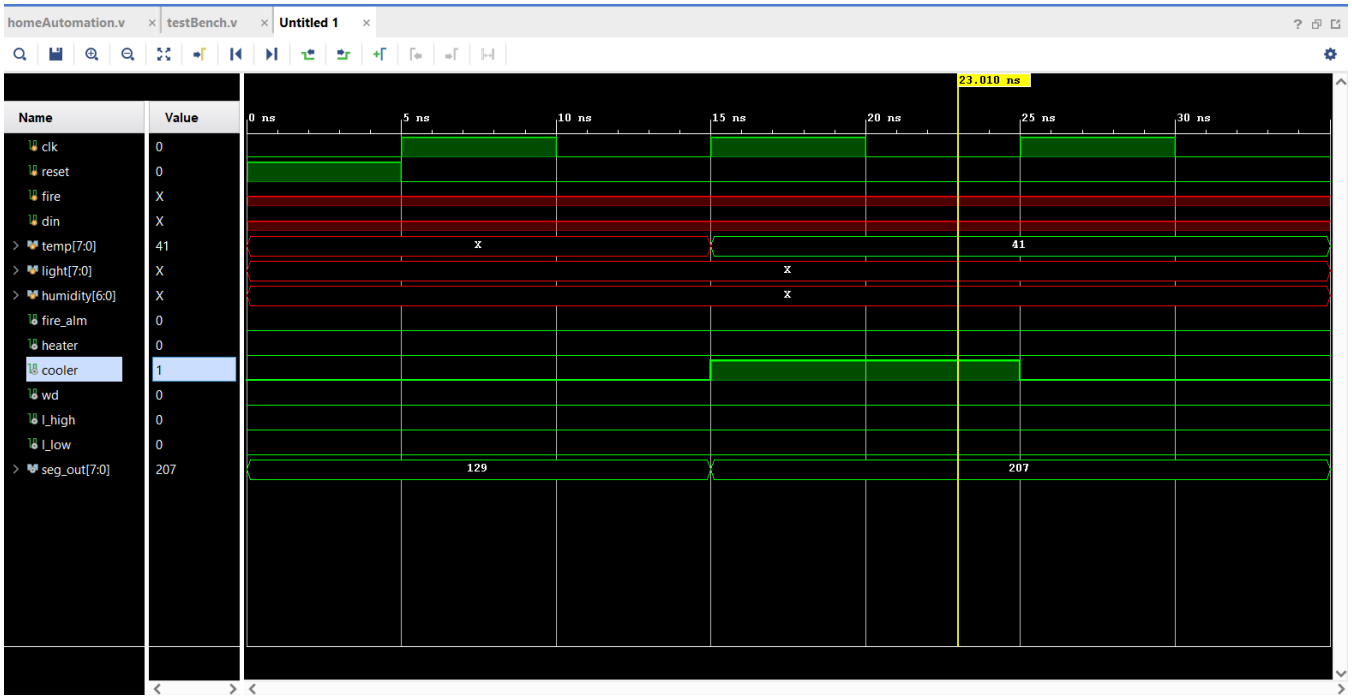
1. **Fire Alarm** :

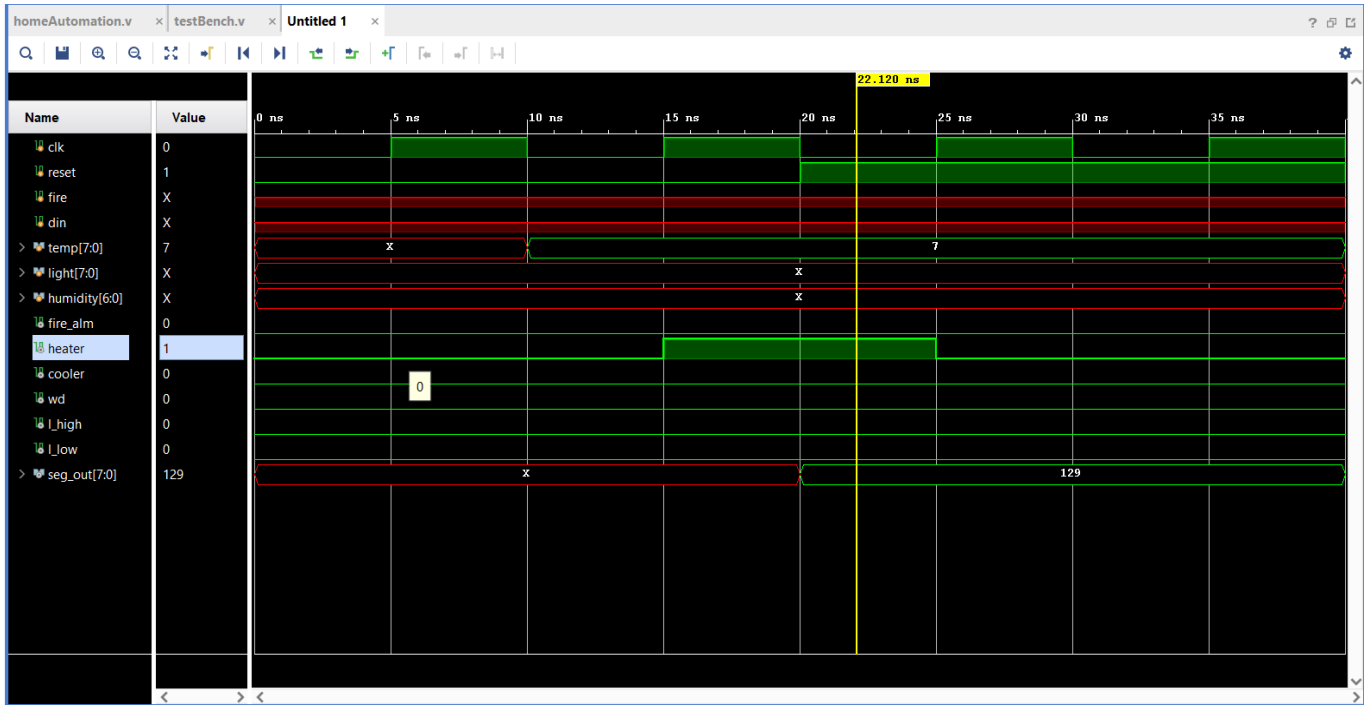
When the smoke detector detects smoke ,it sends the signal ‘fire=1’. And then the fire alarm buzzes if fire is ‘1’.



2. **Temperature Control** :

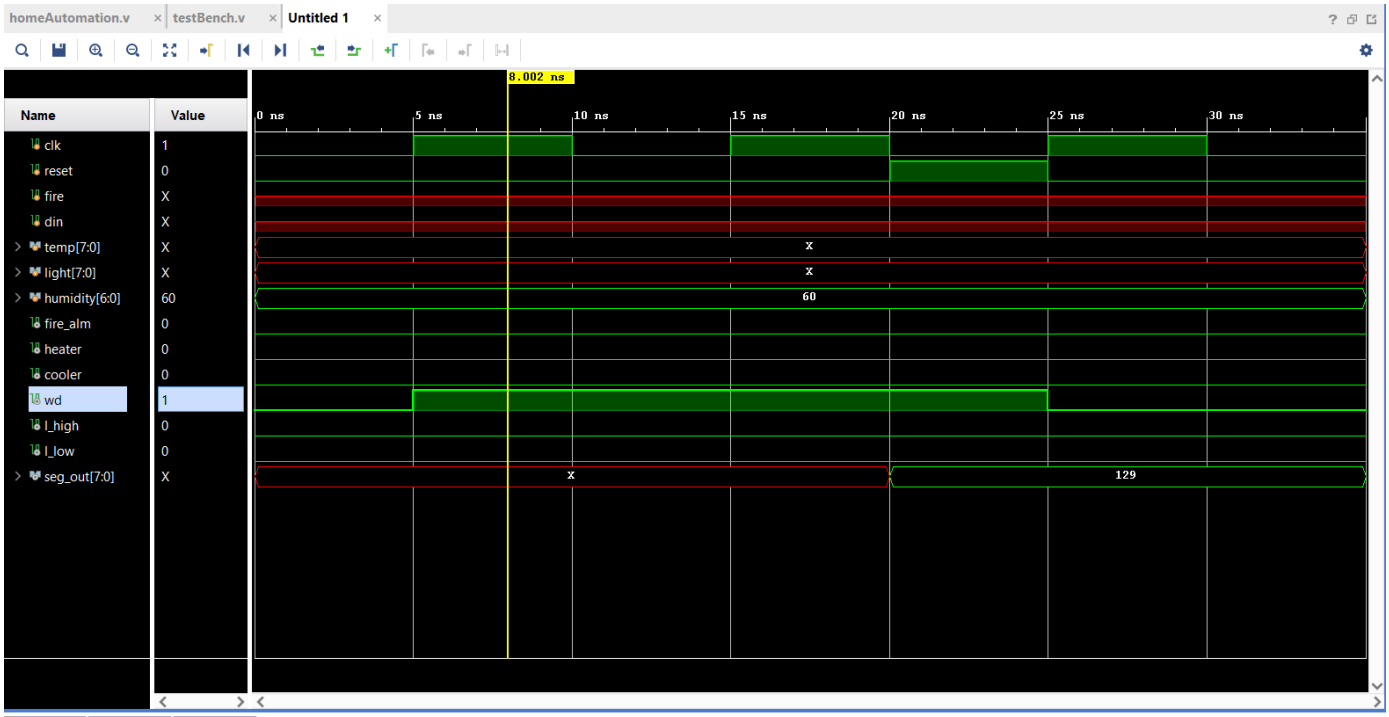
Temperature is detected using a Thermostat. If the temperature is greater than 30 degrees celsius then the Air Conditioner is turned on (cooler =1). If the temperature is less than 15 degrees celsius then Heater is turned on (heater =1).





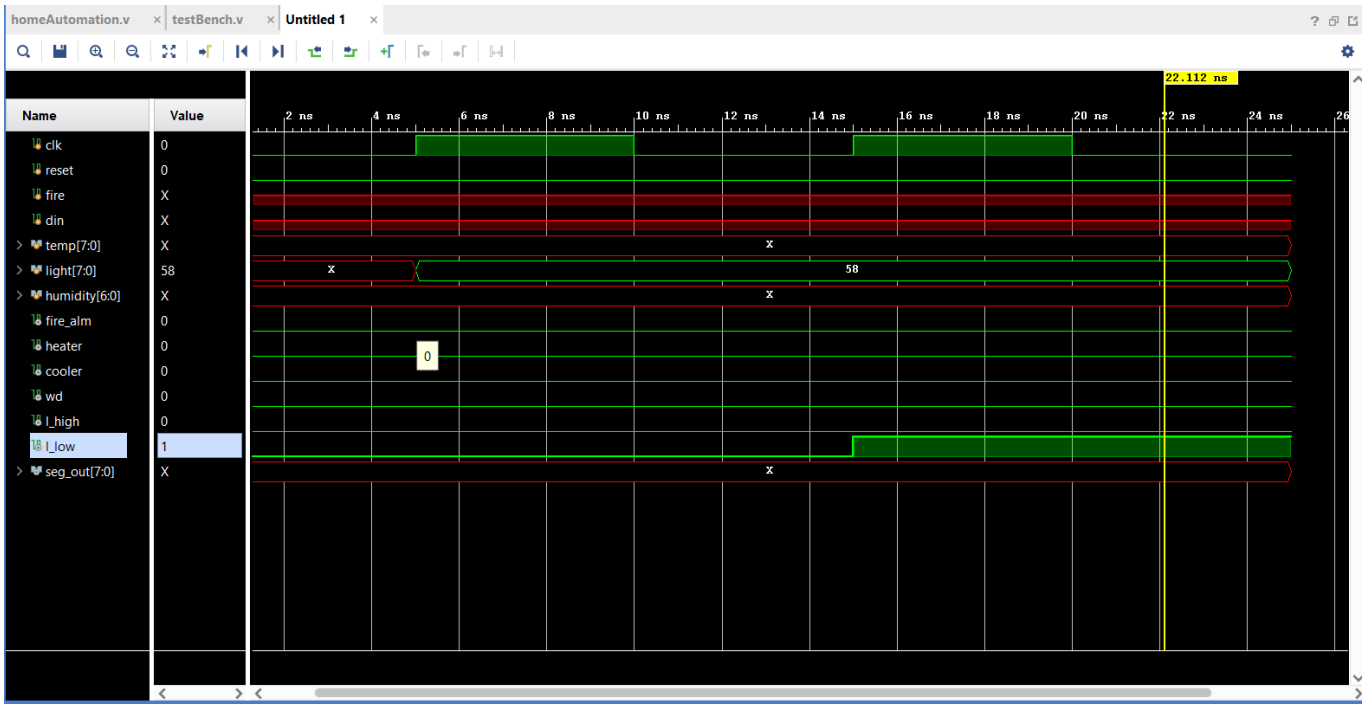
3. **Humidity Control** :

The sensor in this state is the humidity sensor, it has two conditions to fulfill depending on the input it receives. If the humidity is greater than 50 % then open the windows(wd=1).



4. **Luminosity Control** :

Light Intensity is detected by the light-dependent resistor which keeps checking the light intensity in the room. If the light intensity(light) goes below 50 lux or light < 0110010 then increase the current through the light source (l_high=1) and if the light intensity goes above 50 lux or if light > 0110010 then decrease the current through the light source(l_low = 1).



Overall Simulation :

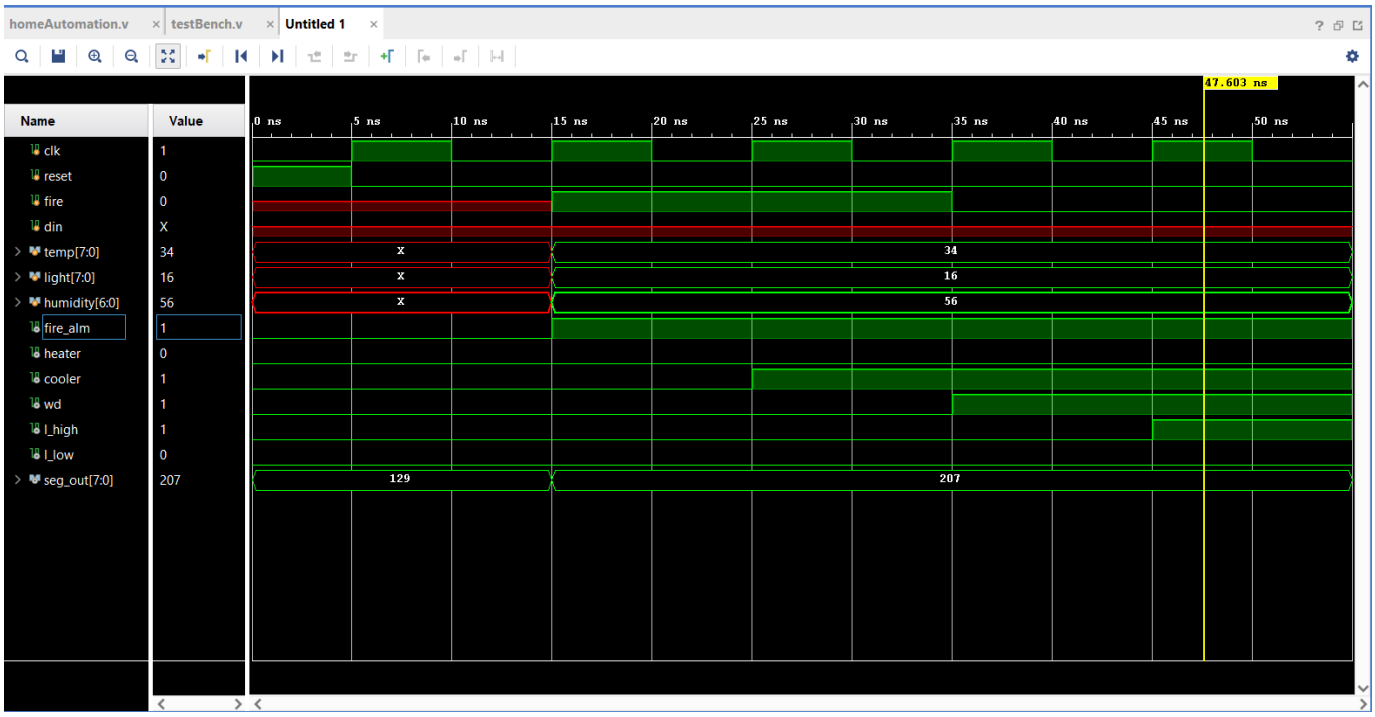
Only for the purpose of simulation let us consider these combination of inputs.

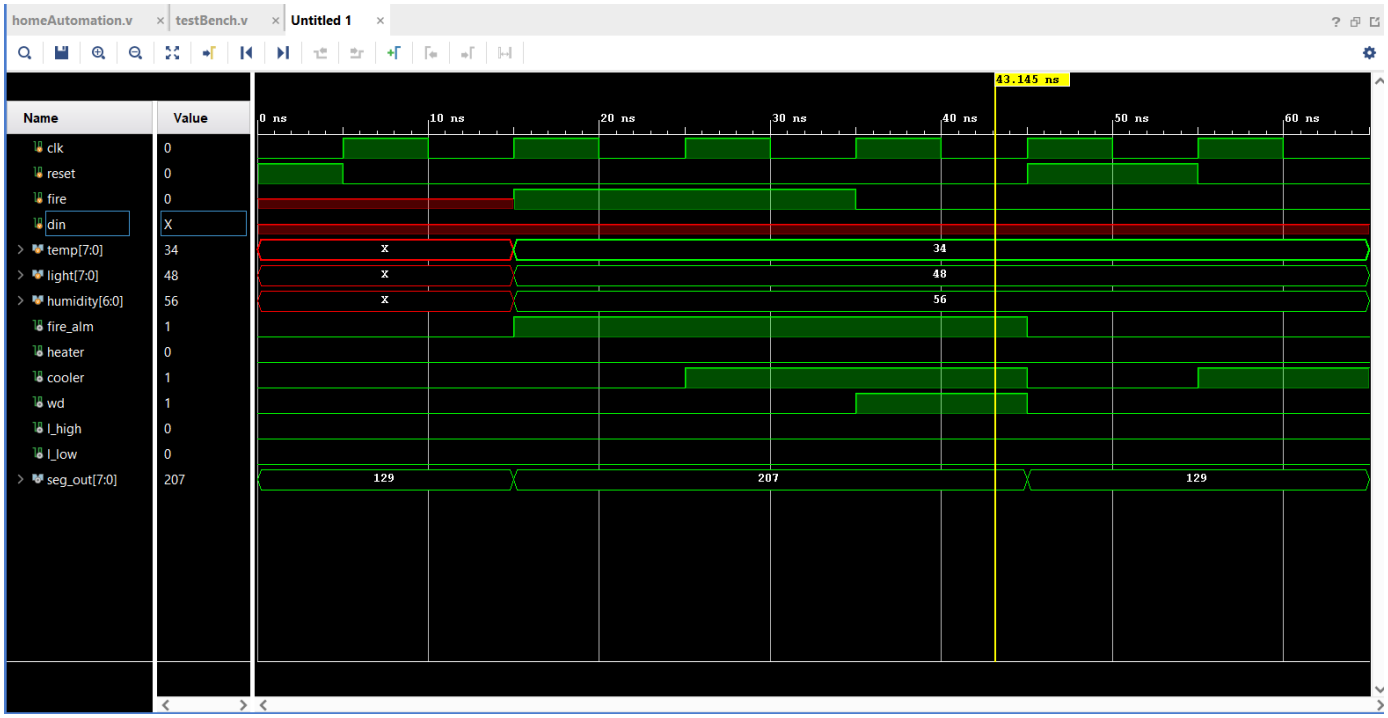
Test Bench for the below simulation :

```
`timescale 1ns / 1ps
module testBench;
reg clk,reset,fire,din;
reg [7:0] temp,light;
reg [6:0] humidity;
wire fire_alm,heater,cooler,wd,l_high,l_low;
wire [7:0] seg_out;
homeAutomation m(.clk(clk),.reset(reset),.fire(fire),.din(din),.temp(temp),
.light(light),.seg_out(seg_out),.humidity(humidity),.fire_alm(fire_alm),
.heater(heater),.cooler(cooler),.wd(wd),.l_high(l_high),.l_low(l_low));
initial clk=0;
always #5 clk = ~clk;
initial
begin
reset=1;
#5 reset=0;
#10 fire=1;temp=8'b00100010;humidity=7'b0111000;light=8'b00010000;
```

```
#20 fire=0;
#20 $finish;

end
endmodule
```





Conclusion :

Automation reduces human effort and increases security . Home automation involves introducing a degree of computerized or automatic control over certain electrical and electronic systems in a home.Although there have been some assumptions, but it is made as close as possible to real life situations. The order of checking the devices based on the priority is followed and also the display shows the state the system is in. Our code is implemented using Verilog HDL in Xilinx. Behavioral style of modeling was used to module the code. Verilog HDL has really helped since it not only combines the hardware and software part, it also provides informative graphs and waveforms which are helpful in understanding the real concept of the project.

Different output waveforms are shown above. As shown in the output waveform; when fire = 1, temperature = 34 degrees celsius , humidity = 56%, light = 16; the respective outputs are fire_alarm=1(from 1st clock) , cooler=1(from 2nd clock), wd=1 (from 3rd clock) , l_high = 1 (from 4th clock) in a sequential way .

References :

Verilog HDL A guide to Digital Design and Synthesis by Samir Palnitkar - for Verilog Language basics and syntax

https://www.youtube.com/playlist?list=PLUtfVcb-iqn-EkuBs3arreilxa2UKIChl - NPTEL Youtube Playlist